Vol.2011-CE-109 No.13
2011/3/19

Java プログラムの起動とそのデータ保存が可能な Wiki システムの試作

山之上　卓†, 山本史弥, 小田謙太郎, 下園幸一

概要：様々な Java プログラムの起動とそのデータ保存を可能とする Wiki システムを試作したことについて述べる．このシステムは PukiWiki と PukiWiki から Java プログラムを起動するためのプラグインと Java のクラスなどで構成される．デフォルト権限の Java Applet は、ローカルにデータを保存することを許可されていない．本システムでは、遠隔でのデータ入出力を容易かつ統一的に行うため Applet のデータ入出力 API を構築した。その上で、保存されるデータをオープンにアクセス可能な Wiki ページとして保存し，Applet の起動のための Wiki タグを導入することより、Wiki システムとの自然な統合を行った．テキストデータの読み書きをする Java プログラムであれば，わずかに書き換えてコンパイルし，クラスファイルを特定のディレクトリに配置するだけで，その Java プログラムを PukiWiki から利用することが可能になる．このとき，PukiWiki のプラグインなどを新たに追加したり，PukiWiki の PHP プログラムを書き換えたりする必要はない．本システムを使うことにより様々な Java プログラムとそのデータが，ネットワーク上で共有可能となる．

Experimental Implementation of a Wiki software which enables to start Java programs and to save their data

Takashi Yamanoue†, Fumiya Yamamoto,
Kentaro Oda, Koichi Shimozono

Abstract : Experimental implementation of a wiki software, which enables to start Java programs and to save their data, is shown. This software consists of PukiWiki which is a wiki software, the plug in which starts Java programs and classes of Java. A Java Applet with default access privilege can't save its data at the local host. We constructed the API of Applets for easy and unified data input and output at a remote host. We also combined the API and the Wiki system by introducing a Wiki tag for starting up Java Applets.

† 鹿児島大学

1. はじめに

学習者にとって魅力的な教育を行うため Java を使ったマルチメディア教材が多く作成されている．Java Applet を使った，Web 上で利用できるマルチメディア教材やプログラミング環境も構築されている [1)2)]．このような環境で作成されたマルチメディアデータやプログラムを，Web 上で授業の参加者やグループで共有し，書き換えることが可能になると，グループ学習などでより有効的にこれらが利用できるようになり，教育の幅が広がる可能性がある．

Wiki は web ページの上でページの編集ができ，共同作業や情報共有の手段として有効であり，Wikipedia のような影響力の大きなサービスでも利用されている．情報処理学会 CE 研究会を含む数多くの学会活動でも Wiki が使われている [10)]．日本で良く使われている wiki クローンの一つとして PukiWiki[9)]がある．実際の授業や学習の現場で PukiWiki を使っているとき，文書と同様に，絵や音楽やプログラムや他の様々なアプリケーションプログラムとそのデータも共有したいと思うことがしばしば生ずる．

我々は，ドロープログラムを PukiWiki で利用できるようにするプラグインと，そのプラグインとお絵かきプログラムを連携するためのアプレットを作成し，NetDraw と名付けている [3)]．今回，ドロープログラムだけでなく，作曲して演奏するための簡単な環境や，プログラム編集しそれを実行する環境など，様々な Java プログラムの起動とそのデータ保存を可能とする Wiki システム，PukiwikiJavaConnector を試作した．このシステムは PukiWiki と PukiWiki から Java プログラムを起動するためのプラグインと Java のクラスなどで構成される．データを文字列として読み書きできる Java プログラムであれば，わずかに書き換えてコンパイルし，クラスファイルを特定のディレクトリに配置するだけで，その Java プログラムを PukiWiki から利用することが可能になる．このとき，PukiWiki のプラグインなどを新たに追加したり，PukiWiki の PHP プログラムを書き換えたりする必要はない．本システムを使うことにより様々な Java プログラムとそのデータが，ネットワーク上で共有可能となる．

2. PukiwikiJavaConnector の利用方法

2.1 インストール

PukiwikiJavaConnector を簡単に利用できるようにするため，Pukiwiki に PukiwikiJavaConnector を組み込んだファイル(pukiwiki-java.tar.gz)を用意している．このファイルをダウンロードし，通常の Pukiwiki と同様にインストールすれば，PukiwikiJavaConnector が使える Pukiwiki が構築できる．

既存の PukiWiki に PukiwikiJavaConnector を加えるためには, PukiwikiJavaConnector のファイル(javaApplications.jar)をダウンロードして解凍し, 解凍された javaApplications ディレクトリとその中身を Pukiwiki のディレクトリの直下にコピーする．使用する PukiWiki のトップディレクトリを <pukiwiki>で表すと, <pukiwiki>/javaApplications となる．次に, <pukiwiki>/plugin ディレクトリに, ブロック型プラグイン jcon.inc.php を追加する．jcon.inc.php の内容をプログラムリスト 1 に示す.

2.1 例題アプリケーションの利用方法

PukiwikiJavaConnector は例題として，簡単なテキストエディタ，簡単な音楽データ編集とその演奏環境，お絵かきプログラム(NetDraw)，ベーシックに類似したプログラミング言語の簡単なプログラム編集・実行環境を持っている．Pukiwiki の編集ページで左端からブロック型プラグイン呼び出し

#jcon(<Java アプリケーション名>)

を記述することにより, そのページを表示したときに<Java アプリケーション名>で指定された Java プログラムが起動される．そのプログラムが読み込んだり, 保存したりするデータは，このページの, プラグイン呼び出しの次の行から記述される.

- **簡単なテキストエディタの利用方法**

簡単なテキストエディタを使う場合, 図 1 のように, PukiwikiJavaConnector の Pukiwiki の編集ページにおいて左端に

#jcon(myEditor)

を記述する. このページを更新してしばらくすると, 図のように３つのウィンドウが新たに表示される.

図 2 において, 左上に表示されている横に細長いウィンドウは, テキストエディタで作成・編集した内容を保存するときに使う「SaveButtonFrame」である．その下の,「myTextArea」と表示されているウィンドウが Java の JTextArea クラスを使って作成した簡単なテキストエディタである．テキストエディタの下に表示されているウィンドウは, Web ページで Java Applet を実行するときに表示される Java コンソールである.

```
<?php
// PukiWiki - Yet another WikiWikiWeb clone
//
// jcon.inc.php
//              t.yamanoue, 2010
// …

function plugin_jcon_convert()
{
    if (PKWK_READONLY) return ''; // Show nothing
    $args = func_get_args();    //引数
    if (count($args) >= 1) { $aw = array_shift($args);} else { $aw = 'draw';}
    $java_application_name = htmlspecialchars($aw,ENT_QUOTES);
    $ret = ''; //戻り値
    $charset=CONTENT_CHARSET;
    $uri=get_script_uri() ;
    $jcode="application.".$java_application_name.".MyApplet.class";
    $plginname="jcon(".$java_application_name.")";
    $ret = <<<EOD
<div>
<applet codebase="./javaApplications/bin"    code="$jcode"
        archive="lib/commons-codec-1.3.jar,lib/commons-httpclient-3.1.jar,
                                lib/commons-logging-1.1.1.jar"
        width="100" height="100">
 <param name="action" value="$uri" />
 <param name="param1" value="plugin=$plginname" />
 <param name="charset" value="$charset" />
 </div>
EOD;
    return $ret;
}
?>
```

プログラムリスト 1.　jcon.inc.php

Vol.2011-CE-109 No.13
2011/3/19

テキストエディタ上で文章を作成，編集した後，SaveButtonFrame の「save」ボタンをクリックすると，その文章が PukiWiki のページに保存される．次回にそのページを開くと，テキストエディタ上に保存した文章が表示される．

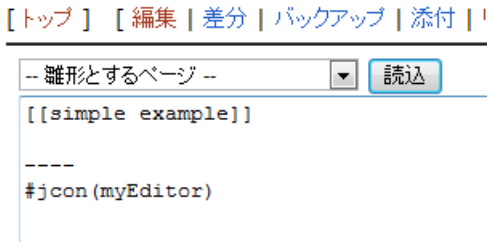

図 1. 編集ページに記述したテキストエディタを起動するブロック型プラグイン

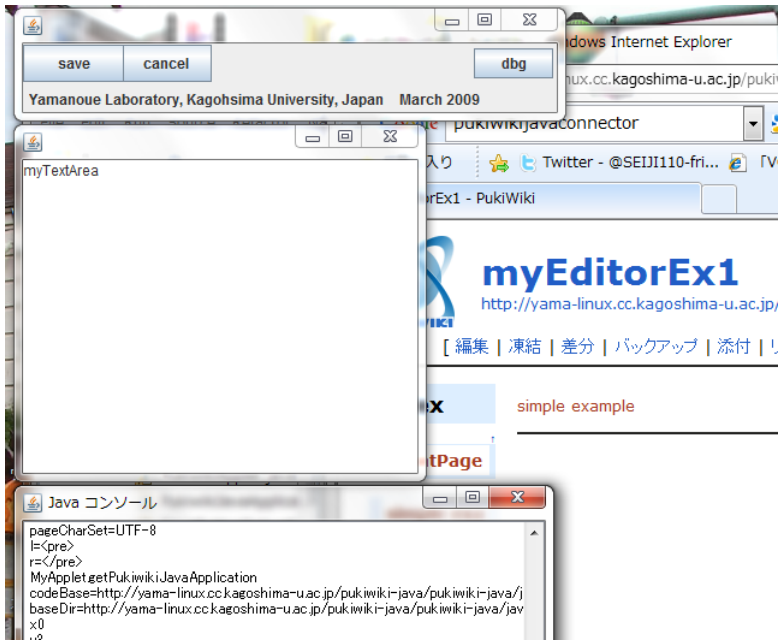

図 2. 起動したテキストエディタと保存用ウィンドウ

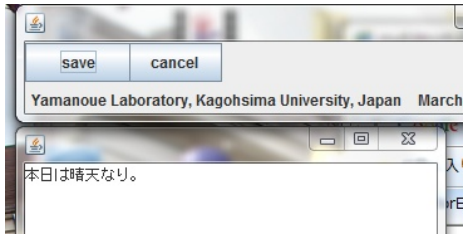

図 3. テキストエディタに書かれた文章とそれが保存された wiki のページ

例えば図 3 のようにテキストエディタ上で「本日は晴天なり」という文章を作成し，save ボタンをクリックするとその文章が PukiWiki ページの #jcon(myTextEditor) の次の行から保存される．PukiWiki の「編集」ボタンをクリックすると，図 3 のように表示され，そのことを確認することができる．

● その他の例題アプリケーションの利用方法

Pukiwiki の編集ページにおいて左端に
#jcon(musicEditor)
を記述する．このページを更新してしばらくすると，図 4 のように音楽編集画面(music editor)と，SaveButtonFrame と Java コンソールが表示される．

音楽編集画面の「メロディー」の欄に，例えば，「3c3c3m3m3,3,3m 」を入力し，再生スピードの欄に「300」を入力し，「play」ボタンをクリックすると「ドドソソララソ」のメロディーが演奏される．音楽編集画面の「save」ボタンか，SaveButtonFrame の save ボタンをクリックすると，テキストエディタのときと同様に，入力したメロディーが，それを表す文字の列として保存される．再度このページが表示されたとき，音楽編集画面のメロディー欄に，保存したメロディーが表れ，play ボタンをクリックするとそのメロディーが演奏される．

PukiWiki の編集ページにおいて左端に
#jcon(draw)

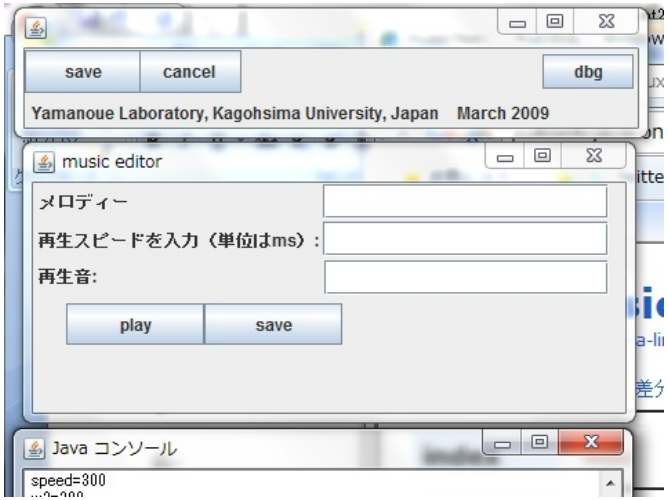

図 4. 簡単な音楽編集・演奏プログラム

を書いて更新すると，お絵かきプログラム(NetDraw) が起動する．

#jcon(basic)

を書いて更新すると，Basic に類似したプログラミング言語(basic)の作成・実行環境が起動する．NetDraw も basic も，作成したデータを保存する方法は，テキストエディタの場合と同じである．図 5 に basic のプログラムを実行した例を示す．

PukiwikiJavaConnector は Basic 認証にも対応している．

3. 新しい Java プログラムの追加

データの保存･読み込みがテキスト形式で行うことができれば，そのような既存の Java プログラムを少し書き換えてコンパイルし，クラスファイルを特定のディレクトリに格納することにより，PukiWiki のページからそのプログラムを起動することが可能になる．このとき，Pukiwiki の php プログラムを変更する必要はない．一種の Factory Method Pattern[5] を使うことにより，新たなプログラムを追加するときの書き換えを少なくしている．図 6 に NewAppli という名前の Java プログラムを追加する場合の，UML のクラス図を示す．新しい Java プログラムを追加する場合以下のように行う．

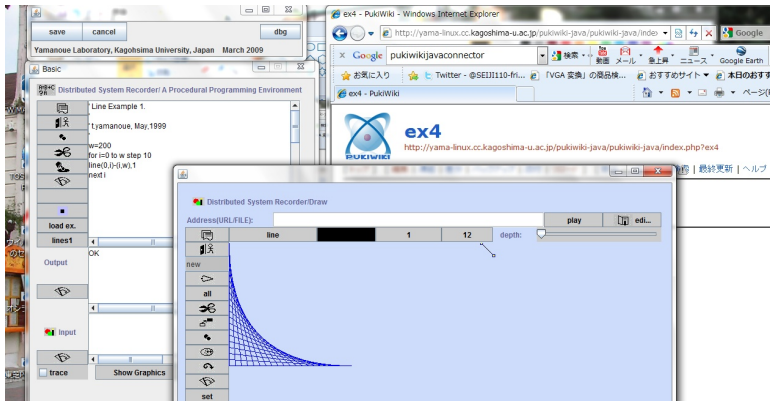

図 5. Basic に類似したプログラミング言語の作成・実行環境

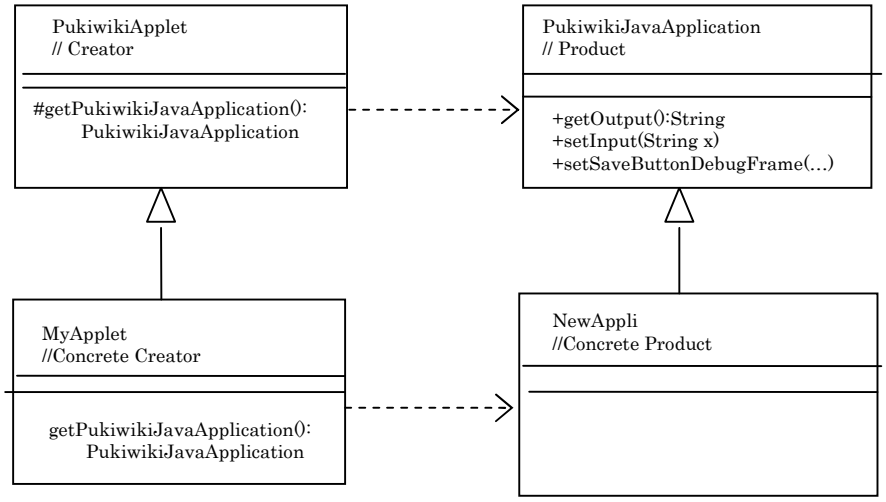

図 6. NewAppli を追加する場合のクラス図

3.1 ソースディレクトリの作成

pukiwiki-java ディレクトリの内に，javaApplications というディレクトリがある．この中に，src という，java のソースプログラムを格納したディレクトリがある．新規に application を追加する場合は，src の中の application ディレクトリの中に，新しいディレクトリを作成する．このディレクトリの名前がブロック型プラグイン jcon の引数となる．例として，この名前を newAppli とした場合，この application は，PukiWiki の編集ページの左端で

#jcon(newAppli)

を書くことにより，起動されるようになる．また，このディレクトリは，パッケージ application.newAppli を表すことになる．以下，newAppli の場合で説明する．

3.2 既存の MyApplet.java のコピーと変更

newAppli ディレクトリに，application.myEditor パッケージ（src の下の application の下の myEditor ディレクトリ）内の，MyApplet.java をコピーする．コピー後は

<pukiwiki>/javaApplications/src/application/newAppli/MyApplet.java

のようになる．MyApplet.java のパッケージ宣言を application.newAppli に変更し，getPukiwikiJavaApplication メソッド内の new MyEditor() を new NewAppli() に書き換える．コンストラクタの引数は，追加する java application に応じて追加する．getPukiwikiJavaApplication メソッドが factory method である．

```
package application.newAppli;
public class MyApplet extends PukiwikiApplet{
   public PukiwikiJavaApplication getPukiwikiJavaApplication(){
      System.out.println("MyApplet.getPukiwikiJavaApplication");
      if(frame==null){
         frame=new MyEditor();
      }
      return this.frame;
   }
}
```

3.3 追加するアプリケーションのソースプログラムのコピー

追加するアプリケーションを newAppli ディレクトリにコピーする．ここで，このアプリケーションは，JFrame 型のクラスを持っている必要がある．このクラスが PukiWiki から起動される．たとえば，このクラス名を NewAppli とする．このプログラムは，NewAppli.java となる．

NewAppli.java をコンパイルするのに必要な他の java ファイルも，このディレクトリなどにコピーする．

3.4 インターフェースの implements

NewAppli.java のパッケージを application.newAppli とし，PukiwikiJavaApplication インターフェースを implements する．

```
package application.newAppli;
…
 public class NewAppli extends JFrame implements PukiwikiJavaApplication
 {
     ...
 }
```

PukiwikiJavaApplication を implements するために必要な，以下の３つの method を NewAppli.java に加える．

```
@Override
public String getOutput() {
        // TODO Auto-generated method stub
}
@Override
public void setInput(String x) {
        // TODO Auto-generated method stub
}
@Override
Public void setSaveButtonDebugFrame
                ( SaveButtonDebugFrame f )  {
        // TODO Auto-generated method stub
}
```

必要に応じて，これらの method の中身を埋める．getOutput() は，PukiWiki のページに保存する NewAppli のデータを取り出す．「save」ボタンがクリックされたとき，この method が呼び出され，NewAppli のデータがこの method を通じて取り出され，Pukiwiki のページに保存される．

setInput(String x) は，Pukiwiki のページから NewAppli が起動された後，そのページに書かれた NewAppli のデータを取り出した後に呼び出される．引数 x が取り出されたデータであり，これを NewAppli で処理することにより，NewAppli にデータが渡されたことになる．setSaveButtonDebugFrame は NewAppli 側で save を呼び出したいときに利用する．

以下に，記述例を示す．

```
public String getOutput() {
        return this.myTextArea.getText(); // add
}
```

```
public void setInput(String x) {
        this.myTextArea.setText(x);    // add
}
public void setSaveButtonDebugFrame
        (SaveButtonDebugFrame f) {
}
```

3.5 コンパイルとコピー

NewAppli ディレクトリ内の Java ファイルをコンパイルし <javaApplications>/bin/application/newAppli ディレクトリを作成し，コンパイルで生成されたクラスファイル MyApplet.class, NewApppli.class および，他の class ファイルを，bin/application/newAppli にコピーする．

なお，本システムは apache の httpclient[13] を利用しており，<javaApplications>/bin/lib に httpclient に関する jar ファイルが必要になる．

4. 評価

本 Wiki システムを利用することにより，既存の Java プログラムを短時間で Wiki システムの一部として組み込むことができるようになった．NetDraw [3]は，教育支援システム SOLAR-CATS[4] に備わっているお絵かきプログラムを PukiWiki で利用できるようにしたものである．これを開発したとき，NetDraw の図データを PukiWiki のページから抽出するための構文解析部を作成したり，PukiWiki ページにお絵かきのデータを埋め込むための様々な変換プログラムを作成したり，このアプレットを起動するための PHP プログラムを作成したりする必要が生じた．PHP や PukiWiki のプログラムを学習する時間も必要であった．このため NetDraw の開発には 3 週間/人程度の労力が必要であった．今回のシステムでは，これらの処理を行うプログラムがあらかじめ javaApplications パッケージの中に組み込まれており，Factory method パターンを利用してこれらを簡単に利用することができる．また，PukiWiki 側のプログラムを書き換える必要がないため，同じプログラムを本システムに組込むのに必要な時間は 3 日/人程度であった．簡単なテキストエディタの場合は 2 時間/人程度，プログラミング言語実行環境と音楽編集・演奏プログラムの場合は 1 日/人程度であった.このとき，Java の開発環境として Eclipse を利用した．

5. 関連研究

九州産業大学藤田研究室の Java アプレットプラグインは任意の Java アプレットを PukiWiki のページに埋め込むことを可能にしている [11]．しかしながら，アプレット上で作成されたデータを保存したり，保存したデータを読み込んだりする機能は持っていない．PukiWiki の上で，共同でお絵かきできるシステムとして「paint」プラグインがある．paint プラグインはペイント専用システムであり，一般的な Java プログラムを利用することはできない．また，回路や設計図などの線画を編集するには不向きである．作曲を共同で行うことを可能にする Web サイトも存在している [8]．しかしながら，このサイトは専用サイトであり，他のサーバで利用することはできない．本システムのように，他の様々なアプリケーションを自由に利用することもできない．

6. おわりに

PukiWiki のページから様々な Java プログラムを起動すると同時に，そのデータを PukiWiki 上で共有可能にするシステムについて述べた．今後多くの利用者が得られるよう，改良し，利用方法のマニュアルの整備などを行っていく予定である．本システムを利用して，既存の教育用アプリケーションとそのデータの共有を可能にできると幸いである．また，教育現場で実際に利用し，その評価を行う予定である．

参考文献

1) 兼宗進, 中谷多哉子, 御手洗理英, 福井真吾, 久野靖, “初中等教育におけるオブジェクト指向プログラミングの実践と評価”, 情報処理学会論文誌:プログラミング, Vol.44, No.SIG13(PRO 18), pp.58-71, Oct. 2003.
2) 西田知博, 原田章, 中村亮太, 宮本友介, 松浦敏雄, “初学者用プログラミング学習環境 PEN の実装と評価”, 情報処理学会論文誌, Vol.48, No.8, pp.2736-2747,Aug.2007.
3) 山之上　卓, “Pukiwiki 用ドロープログラムのプラグインの試作と教育への応用,” 情報教育シンポジウム論文集, 情報処理学会シンポジウムシリーズ vol.2009, No.6, (IPSJ SIGCE SSS2009), pp.55-60, Aug. 2009.
4) Takashi Yamanoue, "A Casual Teaching Tool for Large Size Computer Laboratories and Small Size Seminar Classes" , Proceedings of the 37th annual ACM SIGUCCS conference on User services, pp.211-216, St.Louis, Missouri, US.. 11-14 Oct. 2009.
5) 結城浩, “Java 言語で学ぶデザインパターン入門”, ソフトバンククリエイティブ, 2001.
6) オンライン版ドリトル, http://kanemune.eplang.jp/diary/2009-07-12-1.html
7) PEN のアプレット版, http://kanemune.eplang.jp/diary/2009-07-12-1.html
8) Youself Music, http://yourselfmusic.jp/
9) PukiWiki 公式サイト, http://pukiwiki.sourceforge.jp/
10) 情報処理学会　コンピュータと教育研究会, http://ce.eplang.jp/
11) 九州産業大学藤田研究室, http://w3fj.te.kyusan-u.ac.jp/miwiki/
12) Xampp for windows, http://www.apachefriends.org/jp/xampp-windows.html
13) Http Client Home, http://hc.apache.org/httpclient-3.x/
14) NetDraw, http://yama-linux.cc.kagoshima-u.ac.jp/netdraw